DUPLICATE

Forecasting Research

Forecasting Research Division
Technical Report No. 83

Testing advection schemes for the shallow water equations on a set of test problems I.

by

A.J. Malcolm
March 1994

ORGS UKMO F

National Meteorological Library
FitzRoy Road, Exeter, Devon. EX1 3PB

Meteorological Office
London Road
Bracknell
Berkshire
RG12 2SZ
United Kingdom

Testing advection schemes for the shallow water equations on a set of Test problems, I [1]

A.J. Malcolm

Interim Report

Department of Mathematics

P.O.Box 220

University of Reading

Whiteknights

Reading

RG6 2AX

United Kingdom

[1]The work reported here forms part of the research programme of the Oxford/Reading Institute for Computational Fluid Dynamics and was supported by the Met. Office, Bracknell under contract MET 2a/0471.

1 Introduction

This is an interim report on the numerical study of a conservative finite difference scheme, which is employed in the numerical solution of a climate/forecast model known as the unified model (version 2.8), as set out in Met Office Documentation Paper 10 (1993). This paper is hereafter referred to as DP10.

The scheme is to be validated on the test cases set out in the paper by Williamson et al. (1992) which deals with the treatment of the shallow water equations, hereafter referred to as SWE, in spherical geometry. The final target is to run all the test cases. This report deals with the first test case and outlines the problems encountered in its treatment.

The form of the report consists of the basic spherical geometry identities, a new version of DP10 for the case of the SWE, a summary of DP10 and a discussion of errors and omissions from the original DP10. An explanation of the first test case from Williamson et al. (1992), the problems encountered in applying the scheme from DP10 to the test case, the revised schemes which were actually tested and finally the graphical and numerical results are also included.

The notation in this report will be consistent with DP10 and not with that in Williamson et al. (1992), i.e. spherical coordinates are taken as (λ, ϕ) not (λ, θ).

2 Summary of DP10

DP10 describes the equations and finite difference formulation of the unified forecast/climate model. The finite difference schemes seeks to combine accuracy and efficiency and preserve the conservation properties required for long-term climate integrations. The horizontal grid used is an Arakawa 'B' grid (see Section 3) while a Heun time-step is used for the advection which allows for 2nd or 4th order accuracy in space. In the vertical plane an irregular mesh is used with thin levels near the surface and larger intervals away from the surface. Various quantities

are conserved under the finite difference schemes outlined.

In the first section of DP10, the quantities which are to be modelled and the forecast equations are outlined. The next section deals with the various steps which are are involved in the numerical integration of the equations in time. The different stages of a full integration time step are outlined in more detail in separate subsections, i.e. the adjustment step, the advection step, the treatment of diffusion and the use of Fourier filtering.

The final section of DP10 details how the various quantities are conserved under the finite difference schemes outlined in the previous sections. Finally there are appendices which describe the numerical values which are used for some of the variables in the report when the scheme is implemented in practice.

3 Rewritten Equations

In this section we adapt the equations in DP10 to apply to the SWE on a sphere.

The major differences between these equations and those in DP10 are that there is no vertical velocity and there is only one h level in the vertical, which means that the the k subscripts in DP10 can be ignored. The quantities θ, virtual potential temperature, and Φ, topographic height, do not arise in the SWE. Note that the equations are numbered to be consistent with those in DP10. For clarity the i, j subscripts are suppressed.

Consider then the SWE on the sphere, which are

$$\frac{\partial u}{\partial t} + \frac{u}{a cos\phi}\frac{\partial u}{\partial \lambda} + \frac{v}{a}\frac{\partial u}{\partial \phi} - \left(f + \frac{u}{a}tan\phi\right)v + \frac{g}{a cos\phi}\frac{\partial h}{\partial \lambda} = 0, \tag{5}$$

$$\frac{\partial v}{\partial t} + \frac{u}{a cos\phi}\frac{\partial v}{\partial \lambda} + \frac{v}{a}\frac{\partial v}{\partial \phi} + \left(f + \frac{u}{a}tan\phi\right)u + \frac{g}{a}\frac{\partial h}{\partial \phi} = 0, \tag{6}$$

and

$$\frac{\partial h}{\partial t} + \nabla(h\boldsymbol{u}) = 0, \tag{9}$$

where h is the height, (λ, ϕ) are the spherical coordinates, u, v are the velocities in the coordinate directions respectively, g is gravity and a is the radius of the earth, $\boldsymbol{u} = (u, v)^T$.

The grid used is an Arakawa 'B' grid. The grid axes are in the λ and ϕ directions. The grid is N_λ by N_ϕ points, so $\Delta\lambda = 2\pi/N_\lambda$ and $\Delta\phi = \pi/(N_\phi - 1)$. Thus $\lambda \in [0, 2\pi]$ and $\phi \in [-\pi/2, \pi/2]$, where $\phi = 0$ is the equator, $\phi = \pi/2$ is the North Pole and $\phi = -\pi/2$ is the South Pole. This means that the grid line $j = 1$, with N_λ points, actually represents the North Pole. The Arakawa B grid holds the values of the height field at each grid point. However the velocity field is held at the centre of each grid cell, so the height field is an array of dimension N_λ by N_ϕ, while the velocity field is an array of dimension N_λ by $N_\phi - 1$. The grid sizes can vary but the size used for most of the results in this paper is 96x73. However some simulations are run on grids up to a size of 288x217 points.

The following notation is used :-

$$\delta_\lambda X = \frac{[X(\lambda + \frac{1}{2}\Delta\lambda) - X(\lambda - \frac{1}{2}\Delta\lambda)]}{\Delta\lambda},$$

$$\overline{X}^\lambda = \frac{1}{2}[X(\lambda + \frac{1}{2}\Delta\lambda) + X(\lambda - \frac{1}{2}\Delta\lambda)].$$

Equations (5) and (6) are now split into an adjustment step and an advection step. The advection step solves $\frac{\partial \boldsymbol{u}}{\partial t} = -\boldsymbol{u}.\nabla\boldsymbol{u}$ while the adjustment step solves the rest of equations (5) and (6).

The elements of (5) and (6) which form the adjustment step are manipulated to form a pair of coupled equations, (23) and (24),

$$u^{n+1} = u^n + \delta t \left[\frac{1}{2}F_n(v^{n+1} + v^n) - \frac{g}{a\cos\phi}\overline{\delta_\lambda h^n}^\phi\right], \tag{23}$$

$$v^{n+1} = v^n - \delta t \left[\frac{1}{2}F_n(u^{n+1} + u^n) + \frac{g}{a}\overline{\delta_\phi h^n}^\lambda\right] \tag{24}$$

where

$$F_n = f + \frac{u^n}{a}\tan\phi \qquad \text{and} \qquad f = 2\Omega\sin\phi_A. \tag{25}$$

Eliminating u^{n+1} from (23) and (24) gives

$$v^{n+1} = \frac{v^n\left[1 - \frac{\delta t^2}{4}F_n^2\right] - \frac{\delta t}{2}\left\{2F_n u^n + \frac{2g}{a}\overline{\delta_\phi h^n}^{\lambda} - \frac{g\delta t}{acos\phi}\overline{\delta_\lambda h^n}^{\phi}F_n\right\}}{\left(1 + \frac{\delta t^2}{4}F_n^2\right)}.$$

Similarly, eliminating v^{n+1} from (23) and (24) gives a corresponding expression for u^{n+1}.

Before the remaining terms in (5) and (6) can be employed in the advection step it is necessary to calculate a new height field using (9) which can be solved numerically using the time stepping,

$$h^{n+1} = h^n - \frac{\delta t}{a^2}D, \tag{27}$$

where

$$D = \frac{a}{cos\phi}(\overline{\delta_\lambda(u\overline{h}^{\lambda\phi})}^{\phi} + \overline{\delta_\phi(v\overline{h}^{\lambda\phi}cos\phi)}^{\lambda}) \tag{30}$$

which is a numerical approximation to $\nabla(h\boldsymbol{u})$.

The treatment of the other terms in (5) and (6) can now be implemented during the advection step. In the advection step a new vector $\mathbf{U}$ is defined which is an averaged height weighted velocity field, namely

$$\mathbf{U} = (U, V) = \frac{\sum_{i=1}^{N_A}(au_i\overline{h_i}^{\lambda\phi}, av_i cos\phi\overline{h_i}^{\lambda\phi})}{N_A} \tag{33}$$

where N_A is the number of adjustment steps.

This Heun scheme is then used to produce a new value for u and v which gives the following equations,

$$\begin{aligned} a^2\overline{h^n}^{\lambda\phi}u^{\#} &= a^2\overline{h^n}^{\lambda\phi}u^n - \\ &\frac{\Delta t}{cos\phi}\left((1+\nu)\overline{\overline{U}^{\lambda\phi}}^{\phi}\delta_\lambda u + (1+\nu)\overline{\overline{V}^{\lambda\phi}}^{\lambda}\delta_\phi u - \nu\overline{\overline{U}^{\lambda\phi}}^{3\lambda}\delta_\lambda u - \nu\overline{\overline{V}^{\lambda\phi}}^{3\phi}\delta_\phi u\right) \end{aligned} \tag{37}$$

$$\begin{aligned} a^2\overline{h^n}^{\lambda\phi}v^{\#} &= a^2\overline{h^n}^{\lambda\phi}v^n - \\ &\frac{\Delta t}{cos\phi}\left((1+\nu)\overline{\overline{U}^{\lambda\phi}}^{\phi}\delta_\lambda v + (1+\nu)\overline{\overline{V}^{\lambda\phi}}^{\lambda}\delta_\phi v - \nu\overline{\overline{U}^{\lambda\phi}}^{3\lambda}\delta_\lambda v - \nu\overline{\overline{V}^{\lambda\phi}}^{3\phi}\delta_\phi v\right) \end{aligned} \tag{38}$$

$$a^2\overline{h^{n+1}}^{\lambda\phi}u^{n+1} = a^2\overline{h^{n+1}}^{\lambda\phi}u^n - \frac{\Delta t}{2}\left(\overline{\left(\frac{h^{n+1}}{h^n}\right)}^{\lambda\phi}\underline{U}.\nabla u^n + \underline{U}.\nabla u^{\#}\right) \tag{41}$$

$$a^2\overline{h^{n+1}}^{\lambda\phi}v^{n+1} = a^2\overline{h^{n+1}}^{\lambda\phi}v^n - \frac{\Delta t}{2}\left(\overline{\left(\frac{h^{n+1}}{h^n}\right)}^{\lambda\phi}\underline{U}.\nabla v^n + \underline{U}.\nabla v^{\#}\right) \tag{42}$$

where ν is defined as $\nu = \nu_b(1 - \xi^2_{\max})$. The choice $\nu_b = 0$ defines a 2nd order scheme and $\nu_b = \frac{1}{6}$ defines a 4th order scheme. The minimum value for ν is zero which gives 2nd order accurate space advection. The quantity $\xi_{\max} = \max \xi$ along a line of latitude or longitude as appropriate, and ξ is defined as

$$\xi = \left(\frac{u^2\Delta t^2}{a^2\Delta\lambda^2\cos^2\phi} + \frac{v^2\Delta t^2}{a^2\Delta\phi^2}\right)^{\frac{1}{2}}. \tag{43}$$

If required, diffusion, of order $2j$, of any variable, X, may be added into the scheme using the equations

$$\frac{X^{n+1} - X^n}{\Delta t} = (-1)^{j-1}D^jX, \tag{47}$$

where, for example, if $j = 3, D^3X = D(D(D(X)))$ and

$$D(X) = \frac{1}{a^2h^{n+1}}\left(\frac{1}{cos^2\phi}\delta_\lambda(\overline{k_x(\lambda,\phi)}^{\phi}\delta_\lambda X) + \frac{1}{cos\phi}\delta_\phi(\overline{k_y(\lambda,\phi)}^{\lambda}cos\phi\delta_\phi X)\right)$$

where

$$k_y(\lambda,\phi) = h^{n+1}k \qquad \text{and} \qquad k_x(\lambda,\phi) = k_y(\lambda,\phi)\left(\frac{\Delta\lambda cos\phi}{\Delta\phi}\right)^2. \tag{48}$$

The k in (48) is a user chosen coefficient and for stability we require

$$\Delta t\left(\frac{4k}{a^2\Delta\phi^2}\right)^j \leq 1.$$

These are the main procedures used in a time-step of the numerical algorithm, but there are other procedures mentioned in DP10. The grid splitting mentioned in Section 3.2 of DP10 is not implemented since the use of a 4th order advection scheme renders it redundant. Divergence

damping is also not employed, but Fourier filtering is used to reduce some numerical oscillations introduce by the numerical schemes.

Fourier filtering can be used to reduce the amplitude of unstable wavelengths in the numerical solution. The solution vector is reduced into its component waves each of which has a certain amplitude via FFT. If the solution was reformed and used without modification then numerical oscillations would increase in the solution. A criteria is used to determine which wavelengths are unstable and once the relevant wavelengths are found their amplitude is decreased by reducing the corresponding coefficients, or even setting the amplitude to zero (Fourier chopping). The solution vector is then reconstructed using the modified amplitudes.

4 Implementation over One Time Step

Below is a pseudo-program of what actually occurs in the code during one time-step.

Code Logic for a time-step $u(t) \Rightarrow u(t + \Delta t)$.

Calculation of the filtering criteria is performed once only at the very beginning of the time-step.

Adjustment

$\tau = t, \qquad \delta t = \dfrac{\Delta t}{\text{no of adjustment steps}}$

Do $I = 1, \text{no of adjustment steps}$

Calculate $u(\tau + \delta t), v(\tau + \delta t)$ from Equations (23) and (24)

Fourier filter $h(\tau)u(\tau + \delta t), h(\tau)v(\tau + \delta t)$

Calculate $D(\tau + \delta t)$ from filtered hu, hv from Equation (30)

Calculate $h(\tau + \delta t)$ from Equation (27)

Set $\tau = \tau + \delta t$

End Do

Advection

Calculate $u(t+\Delta t), v(t+\Delta t)$ from Equations (37), (38), (39), (40), (41) and (42), using the final $u(\tau), v(\tau)$ from the adjustment as u^n, v^n in those equations.

Diffusion

Calculate diffusion increments for u and v using Equations (47), (48) and add on.

Fourier filter the $hu(t+\Delta t), hv(t+\Delta t)$ at the end of this step.

This is the completion of one time-step.

5 Problems with DP10

There were a few problems with the model documentation paper.

In Section 3.5 (Fourier filtering) the definition of k in equations (51) and (52) is erroneous and leads to cancellation in the two equations. It should actually just be defined as the wave number running from 1 to $N_\lambda/2$.

The treatment of the poles does not mention that ϕ at the pole is defined as $\pm(\frac{\pi}{2} - \frac{\Delta\phi}{4})$. As a whole the treatment of the polar values for all the variables is unclear from a reading of DP10.

There is also no indication of the ratio of adjustment time-step to advection time-step.

These problems were overcome after discussions with the paper's authors.

6 The Williamson First Test Case

The complete suite of test cases from the paper by Williamson et al. systematically tests areas of the advection code.

The first test case does not deal with the full SWE but solely with the advective part. Full shallow water codes should be able to apply it by overwriting the calculated velocity field with the specified velocity field at all times.

This case is a test of the ability of the numerical scheme to cope with the transport of a cone around the globe. Different orientations of the advecting wind are specified. An initial height field is specified and a velocity field is specified which is analytically divergence free. The advecting wind is given by

$$u = u_0(\cos\phi\cos\alpha + \sin\phi\cos\lambda\sin\alpha), \tag{49}$$

$$v = -u_0\sin\lambda\sin\alpha, \tag{50}$$

where the parameter α is the angle between the axis of solid body rotation and the polar axis of the spherical coordinate system.

The initial height field is a cosine bell given by

$$h(\lambda,\phi) = \begin{cases} \frac{h_0}{2}(1+\cos(\frac{\pi r}{R})) & \text{if } r < R \\ 0 & \text{if } r \geq R \end{cases} \tag{51}$$

where $h_0 = 1000$m, $R = a/3$ and r is the great circle distance between (λ,ϕ) and the centre of the cone, which is initially at $(\lambda_c,\phi_c) = (3\pi/2, 0)$, where

$$r = a\cos^{-1}[\sin\phi_c\sin\phi + \cos\phi_c\cos\phi\cos(\lambda-\lambda_c)]. \tag{52}$$

The initial height field actually used has all the h values increased by 1000m and can be seen in Figure 1. (This addition results in equations (37) to (40) being defined everywhere.)

Various numerical error estimates are also introduced in Williamson et al. (1992) and take two forms. The first type are normalised global errors for a variable in various norms. The second type are normalised means, variances, maxima and minima for a variable.

The suggested form of the graphical output is also outlined in the original paper.

The true solution of the problem is that the cone rotates around the globe without any change in shape. It is possible to calculate the expected centre of the cone at all times, so comparisons between the true solution and the calculated solution are possible.

7 Problems with Original Scheme

The original scheme, for the full SWE, was tried for this problem. The split adjustment/advection scheme used $\nabla(h\boldsymbol{u})$ to calculate the change in h. The actual data is analytically non-divergent in areas where h is constant, but when it is introduced into the numerical scheme the scheme becomes numerically divergent as is shown below.

The velocity field is set analytically as

$$\begin{aligned} u &= u_0(cos\phi cos\alpha + sin\phi cos\lambda sin\alpha), \\ v &= -u_0 sin\lambda sin\alpha. \end{aligned}$$

The analytic divergence is

$$\begin{aligned} \nabla.\mathbf{u} &= \frac{1}{acos\phi}\left(\frac{\partial u}{\partial \lambda} + \frac{\partial (vcos\phi)}{\partial \phi}\right) \\ &= \frac{1}{acos\phi}[-u_0 sin\phi sin\alpha sin\lambda + \frac{\partial}{\partial \phi}(-u_0 sin\lambda sin\alpha cos\phi)] \\ &= \frac{-u_0 sin\alpha sin\lambda}{acos\phi}[-sin\phi + sin\phi] = 0. \end{aligned}$$

The finite difference operator D is defined as

$$D = \frac{a}{cos\phi}(\overline{\delta_\lambda(u\overline{h}^{\lambda\phi})}^{\phi} + \overline{\delta_\phi(v\overline{h}^{\lambda\phi}cos\phi)}^{\lambda}).$$

If h is constant then

$$D = \frac{ah}{cos\phi}(\overline{\delta_\lambda(u)}^{\phi} + \overline{\delta_\phi(vcos\phi)}^{\lambda}),$$

where $\overline{\delta_\lambda(u)}^{\phi}$ has the stencil $\frac{1}{2\Delta\lambda}\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$. $\overline{\delta_\phi(vcos\phi)}^{\lambda}$ has the stencil $\frac{1}{2\Delta\phi}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$.

Centred on an h point D gives for the analytic $\mathbf{u}$ field

$$\overline{\delta_\lambda(u)}^{\phi} = \frac{u_0}{2\Delta\lambda}\begin{bmatrix} cos(\phi + \frac{\Delta\phi}{2})cos\alpha & + & sin(\phi + \frac{\Delta\phi}{2})cos(\lambda + \frac{\Delta\lambda}{2})sin\alpha \\ +cos(\phi - \frac{\Delta\phi}{2})cos\alpha & + & sin(\phi - \frac{\Delta\phi}{2})cos(\lambda + \frac{\Delta\lambda}{2})sin\alpha \\ -cos(\phi + \frac{\Delta\phi}{2})cos\alpha & - & sin(\phi + \frac{\Delta\phi}{2})cos(\lambda - \frac{\Delta\lambda}{2})sin\alpha \\ -cos(\phi - \frac{\Delta\phi}{2})cos\alpha & - & sin(\phi - \frac{\Delta\phi}{2})cos(\lambda - \frac{\Delta\lambda}{2})sin\alpha \end{bmatrix}$$

$$= \frac{u_0 sin\alpha}{2\Delta\lambda}\left(sin(\phi+\frac{\Delta\phi}{2})+sin(\phi-\frac{\Delta\phi}{2})\right)\left(cos(\lambda+\frac{\Delta\lambda}{2})-cos(\lambda-\frac{\Delta\lambda}{2})\right)$$

$$= \frac{u_0 sin\alpha}{2\Delta\lambda}\left(2sin\phi cos\frac{\Delta\phi}{2}\right)\left(-2sin\lambda sin\frac{\Delta\lambda}{2}\right),$$

and

$$\overline{\delta_\phi(vcos\phi)}^{\lambda} = \frac{-u_0 sin\alpha}{2\Delta\phi}\begin{bmatrix} cos(\phi+\frac{\Delta\phi}{2})sin(\lambda+\frac{\Delta\lambda}{2}) & + & cos(\phi+\frac{\Delta\phi}{2})sin(\lambda-\frac{\Delta\lambda}{2}) \\ -cos(\phi-\frac{\Delta\phi}{2})sin(\lambda+\frac{\Delta\lambda}{2}) & - & cos(\phi-\frac{\Delta\phi}{2})sin(\lambda-\frac{\Delta\lambda}{2}) \end{bmatrix}$$

$$= \frac{-u_0 sin\alpha}{2\Delta\phi}\left[cos(\phi+\frac{\Delta\phi}{2})-cos(\phi-\frac{\Delta\phi}{2})\right]\left[sin(\lambda+\frac{\Delta\lambda}{2})+sin(\lambda-\frac{\Delta\lambda}{2})\right]$$

$$= \frac{-u_0 sin\alpha}{2\Delta\phi}\left[-2sin\phi sin\frac{\Delta\phi}{2}\right]\left[2sin\lambda cos\frac{\Delta\lambda}{2}\right].$$

Giving

$$D = \frac{ahu_0 sin\alpha}{2cos\phi}\left(\frac{-4}{\Delta\lambda}sin\phi sin\lambda cos\frac{\Delta\phi}{2}sin\frac{\Delta\lambda}{2}+\frac{4}{\Delta\phi}sin\lambda sin\phi sin\frac{\Delta\phi}{2}cos\frac{\Delta\lambda}{2}\right)$$

$$= 2ahu_0 sin\alpha sin\lambda tan\phi\left(\frac{-1}{\Delta\lambda}cos\frac{\Delta\phi}{2}sin\frac{\Delta\lambda}{2}+\frac{1}{\Delta\phi}sin\frac{\Delta\phi}{2}cos\frac{\Delta\lambda}{2}\right).$$

This result means that when the advecting wind is not around the equator the numerical scheme introduces spurious numerical errors at the poles which filtering was unable to remove and the program failed. In practice, even with an equatorial wind flow there were still numerical oscillations introduced at the poles due to rounding errors.

In this first test case, since the advecting wind was overwritten at each stage of the calculation, the quantity which was diffused was not the velocities but the increment in the height field. This was also the quantity which was filtered.

The numerical non divergence of the analytically divergence free wind field led to a change in solution procedure which is documented in the following section.

8 Testing the Heun Scheme

The numeric non-zero divergence was actually overcome by totally changing the nature of the algorithm. As the height field is the only changing variable it was decided that the only step in the algorithm would be an advection step on the height field using the same Heun scheme as the actual algorithm. Thus the adjustment phase of the algorithm is ignored and the height field, not the velocity field, is advected.

The Heun scheme used was a Modified two stage method which involved the same parameter as for the advection treatment of the velocity in the unified scheme.

Over a full rotation this method did fail for wind fields over the poles but it carried on longer than the previous schemes. With the introduction of filtering and diffusion the scheme did produce results for a full 12 day cycle.

9 TVD Scheme

Another method tested was a TVD scheme for the treatment of advection. This TVD scheme is the one documented in Leathwood and Cullen (1990). It was set up to use either a superbee or Van Leer limiter and both were tested. It was implemented on the test problem and was seen to produce reasonable results even over the pole.

10 Results

In this section the graphical and numerical results are presented.

The results are shown as contour plots for, where applicable and, in order, original scheme, Heun scheme, Heun scheme plus diffusion, TVD (superbee) and TVD (Van Leer).

These plots are produced for different values of α, the advecting wind direction. Also produced are plots of numerical error vs diffusion coefficient, and numerical error vs time-step.

It is possible to see from Figure 2 that the advection/adjustment scheme is poor even for the test of an advecting wind around the equator. The introduction of the pure Heun scheme with no filtering or diffusion improved the representation, see Figure 3, but was plagued by numerical oscillations in the height field behind the cone.

The TVD schemes reduce the oscillations, as expected, see Figure 7, but with a resultant loss in the height field. The superbee limiter produced a reduction to about 75% while the van leer limiter produced a reduction to about 50% of the original maximum height, see Table 1.

For equatorial flow the best results were produced by the Heun scheme together with filtering and diffusion. The filtering reduced the numerical oscillations in the polar regions while the diffusion reduced the numerical oscillations behind the cone. Thus while the actual contours are slightly more spread out the loss in height is not as severe, see Table 1 and Figures 5 and 6.

The equatorial advecting wind allowed a numerical comparison of the effect of choosing different values of the diffusion coefficient, k. This comparison shows that the value of k which produces a best result is slightly below that used in the actual models. This value is 7.5×10^7 while that used in the model is actually 1.2×10^8 or 1.6×10^8 , see Figure 12.

Another set of tests whose results are shown is the effect of reducing the time-step successively by a factor of 2. The values for the errors show that the numerical error tends upwards to a limit as the time-step is halved repeatedly, see Figure 15.

The next set of test results documented is actually that for an advecting wind direction of $\alpha = 15°$. This is far enough away from the equatorial flow to introduce polar effects but not to be totally dominated by them.

In all the methods it was necessary to reduce the time-step by a factor of four, from that suggested in DP10, to get any results at all. Compare Figures 8 and 9.

The best results were produced by the TVD schemes, see Figure 10, while the other schemes

produced skewed cones which were not in the correct position. However even with these poor results it is still possible to do a plot of error against diffusion coefficient. In this case it can be seen that the best value of k is 2.5×10^8, see Figure 13.

The final cases investigated were those which involved flow over the poles. It was decided that to run the code for a full 12 day cycle was too severe a test as any practical problem would involve at most flow over one pole. Thus the codes were run for 6 days rather than 12 days so that the flow should end up on the equator but on the opposite side of the globe.

The only scheme which worked for that length of time without blowing up was the TVD scheme and even in this case the cone was severely distorted, see Figure 14.

Below is a table of some numerical results which can be used to look at different schemes.

96 x 73 grid, $\alpha = 0$, 30 minute time step			
scheme	RMS error	Max height	Min height
Heun scheme	0.2455	1937.5	836.4
Heun scheme +diffusion (4th order)	0.3202	1707.3	955.1
TVD (superbee)	0.2569	1739.5	1000.0
TVD (van leer)	0.4750	1540.0	1000.0
Heun scheme (2 minute time step)	0.2831	1930.3	828.5

Table 1:

A semi-lagrangian code (Bates et al.(1990)) was also implemented and the results for this can be seen in Figures 18, 19 and 20. These show, respectively the contours when the advecting wind is at an angle, $\alpha = 0, \alpha = 15°$ and $\alpha = 90°$. The time step for all these is 30 minutes and surprisingly the best result is that for $\alpha = 90°$.

11 Conclusions

It can be seen that none of the schemes tested are successful in coping with the first test case in spherical geometry. For equatorial flows both the Heun scheme and the TVD with a superbee limiter produce similar results, but neither produces reasonable results. However with a non equatorial advecting wind it can be seen that the TVD scheme is much better at advecting the height field.

It can also be seen that the choice of diffusion coefficient currently in use is acceptable but could perhaps be modified to produce slightly better results.

12 Acknowledgements

I would like to thank the Meteorological Office for the funding of the contract under which I am working and Dr Mike Cullen, Dr Terry Davies and Mark Mawson there for their advice and helpful suggestions. I would also like to thank Dr Mike Baines and Dr Peter Sweby for useful discussions during the course of this work.

References

[1] **S.A. Leathwood and M.J.P. Cullen**, (1990), *A Positive Definite Advection Scheme Suitable for the Unified Model*, Met. Office Short-Range Forecasting Research Technical Note No. **55**

[2] **M.J.P. Cullen, T. Davies and M.H. Mawson**, (1993), *Conservative Finite Difference Schemes for a Unified Forecast/Climate Model*, Met. Office Unified Model Documentation Paper No. 10, Version No. 24

[3] **D.L. Williamson, J.B. Drake, J.J. Hack, R. Jakob and P.N. Swarztrauber**, (1992), *A Standard Test Set for Numerical Approximations to the Shallow Water Equations in Spherical Geometry*, J. Comp. Phys. **102**, pp 211-224

[4] **B.L. Marshall**, (1989), *Tests of the New Heun Advection Scheme*, Met. Office 11 Technical Note No. **31**

[5] **J.R. Bates, F.H.M. Semazzi, R.W. Higgins and S.R.M. Barros**, (1990), *Integration of the shallow-water equations on the sphere using a vector semi-lagrangian scheme with a multigrid solver*, Mon. Wea. Rev. **118**, pp 1615-1627

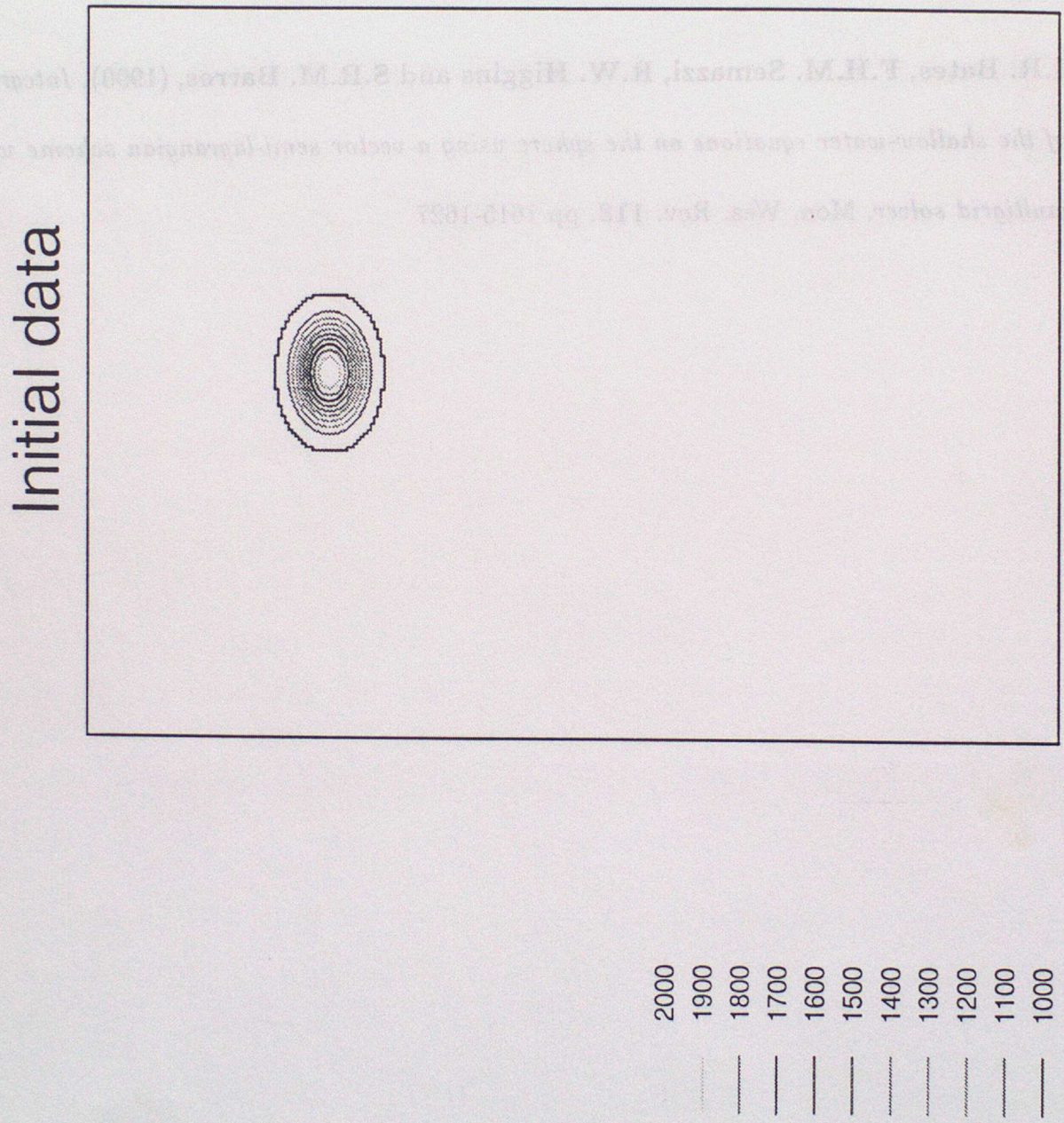


Figure 1: Contour plot of the initial data, a cosine cone

Figure 2: Unified model advection scheme, time step 1800 seconds, $\alpha = 0$, 96 x 73 grid, result after 12 days.

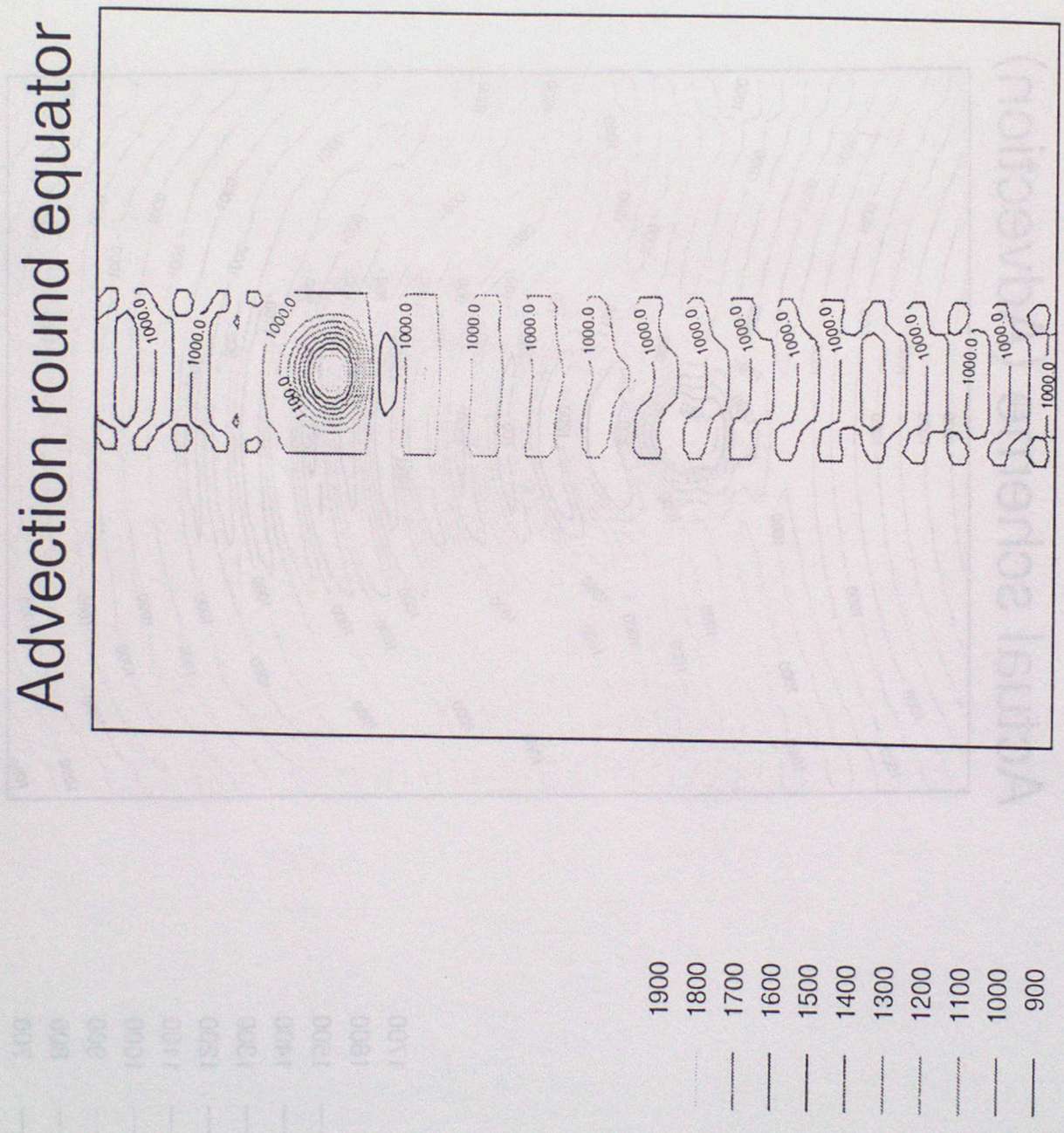


Figure 3: Heun scheme, time step = 1800 seconds, $\alpha = 0{,}96$ x 73 grid, result after 12 days.

Figure 4: Heun scheme, time step = 600 seconds, α = 0,288 x 217 grid, result after 12 days.

Figure 5: Heun scheme with filtering and 4'th order diffusion, time step = 1800 seconds, $\alpha = 0$, $k = 1.6x10^8$, 96 x 73 grid, result after 12 days.

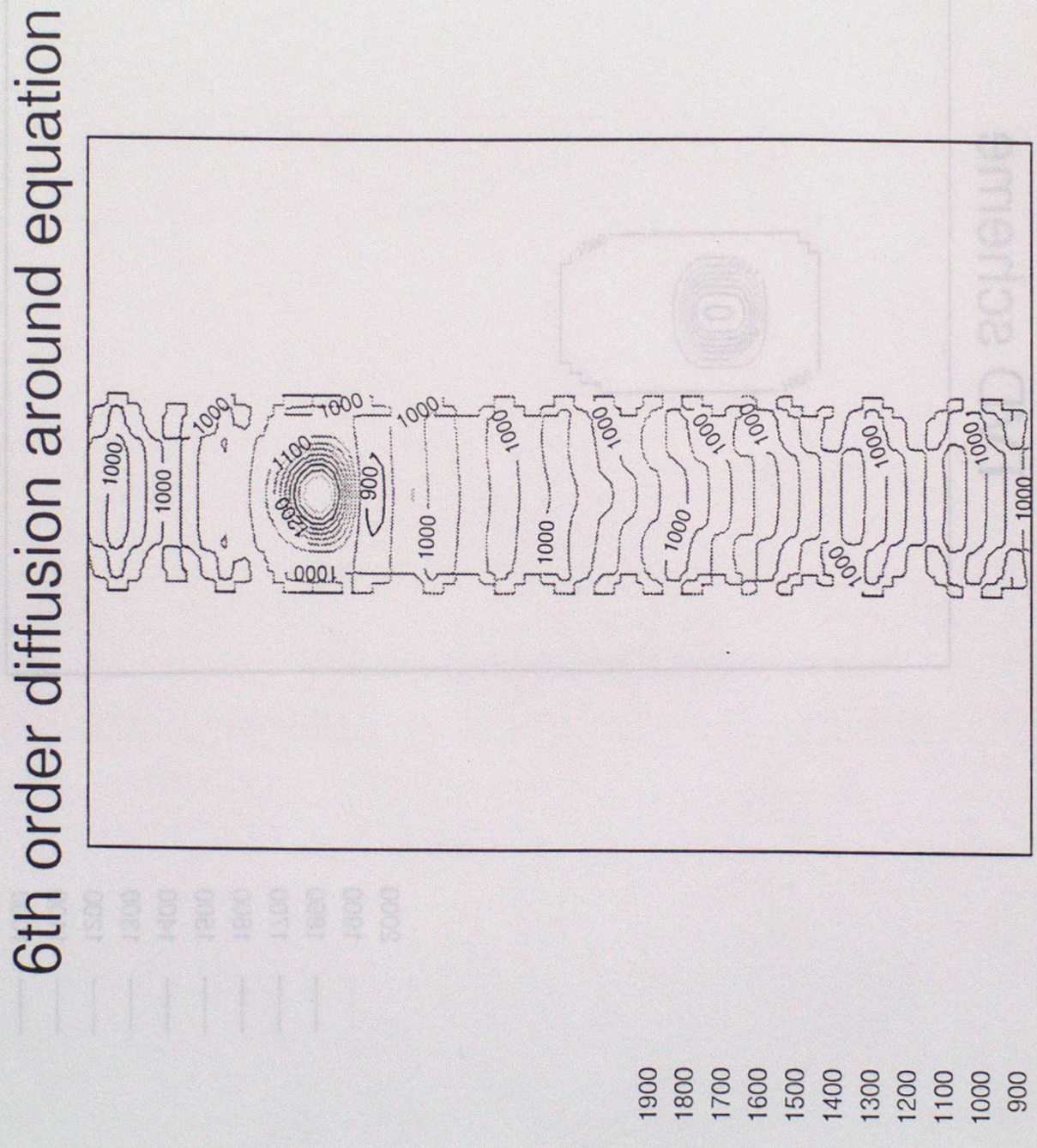


Figure 6: Heun scheme with filtering and 6'th order diffusion, time step = 1800 seconds, $\alpha = 0$, $k = 8x10^8$, 96 x 73 grid, result after 12 days.

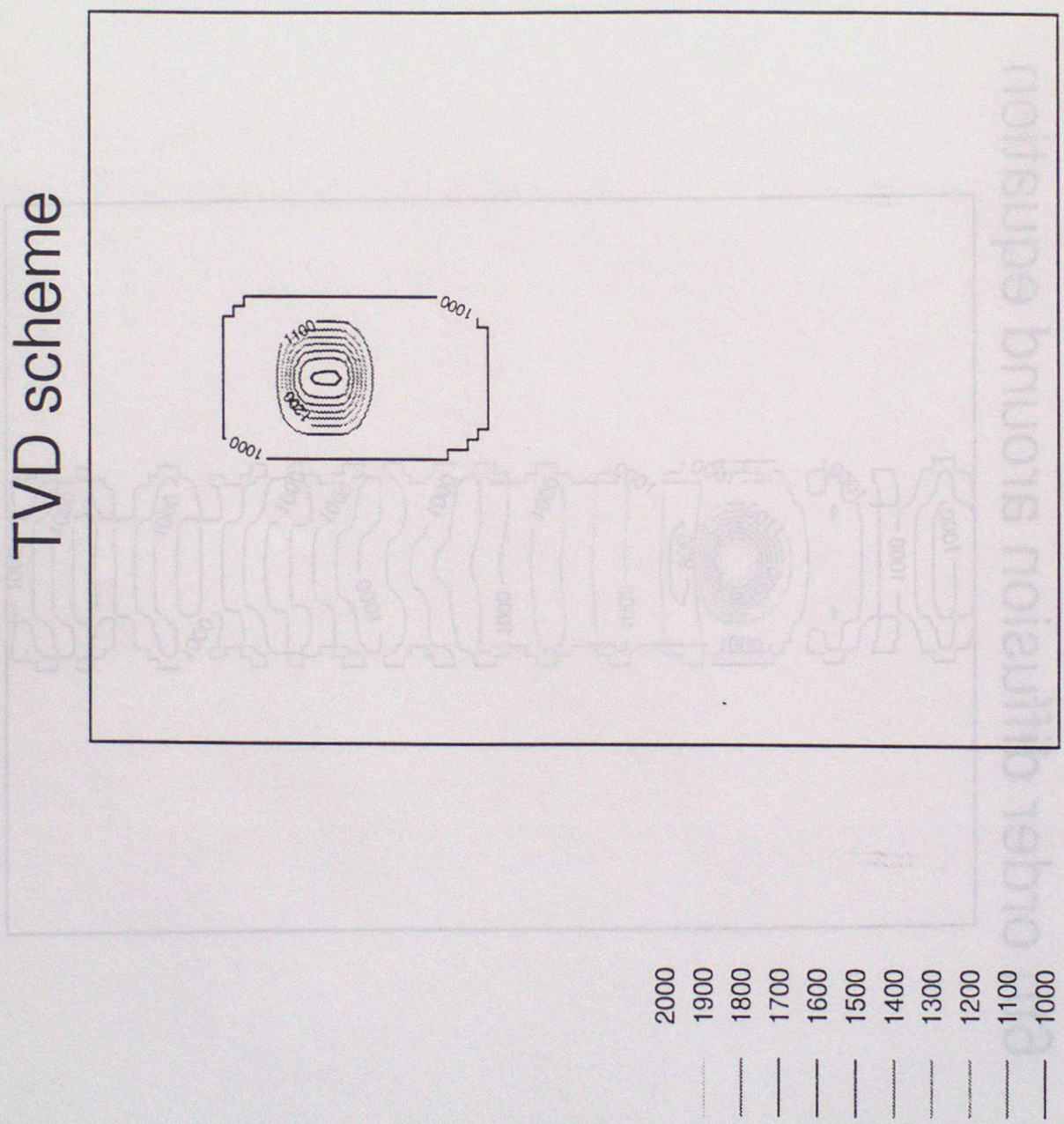


Figure 7: TVD scheme , Superbee limiter, time step = 1800 seconds, $\alpha = 0$, 96 x 73 grid, result after 12 days.

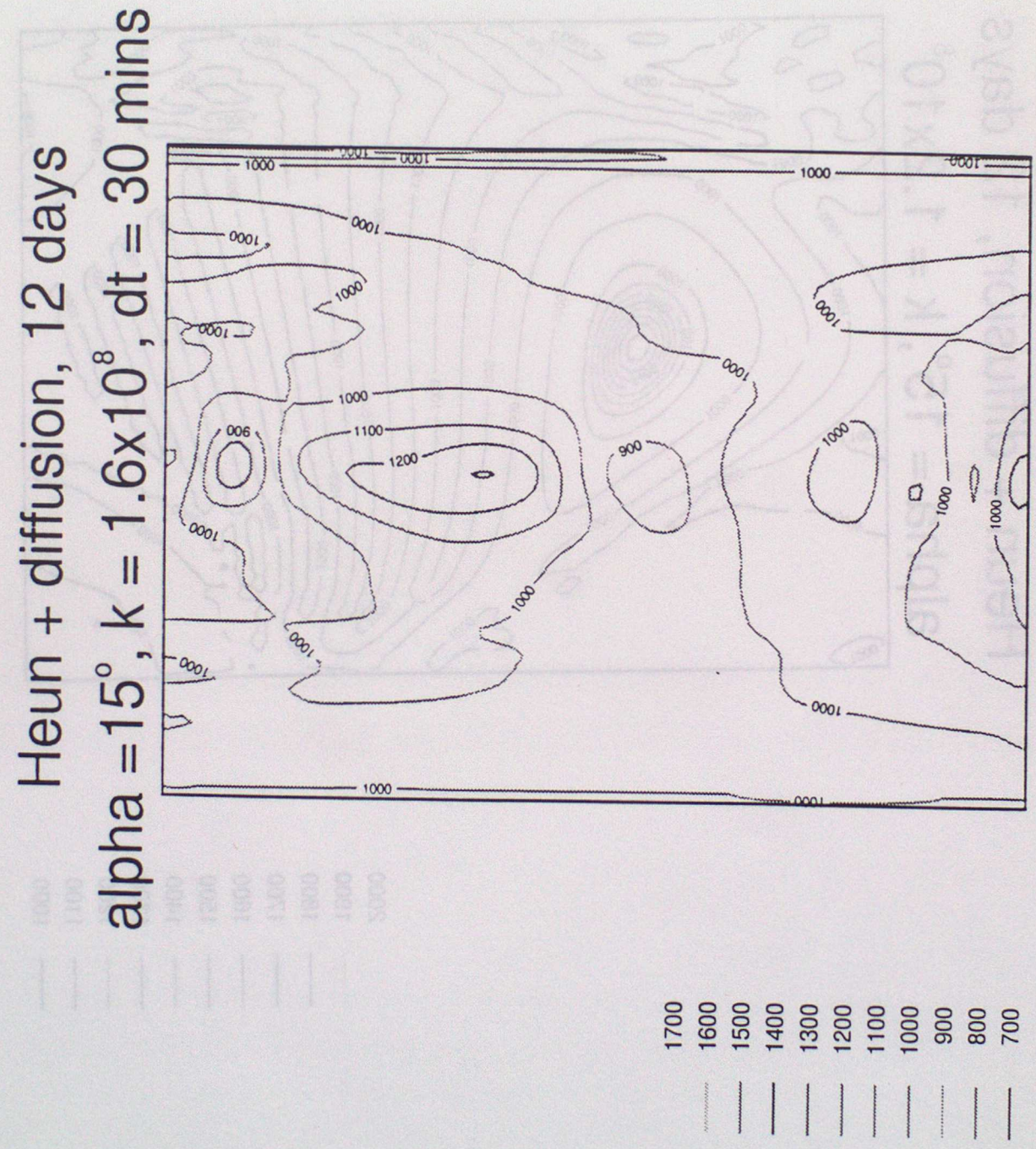


Figure 8: Heun scheme with filtering and 4'th order diffusion, time step = 1800 seconds, $\alpha = 15°$, $k = 1.6x10^8$, 96 x 73 grid, result after 12 days.

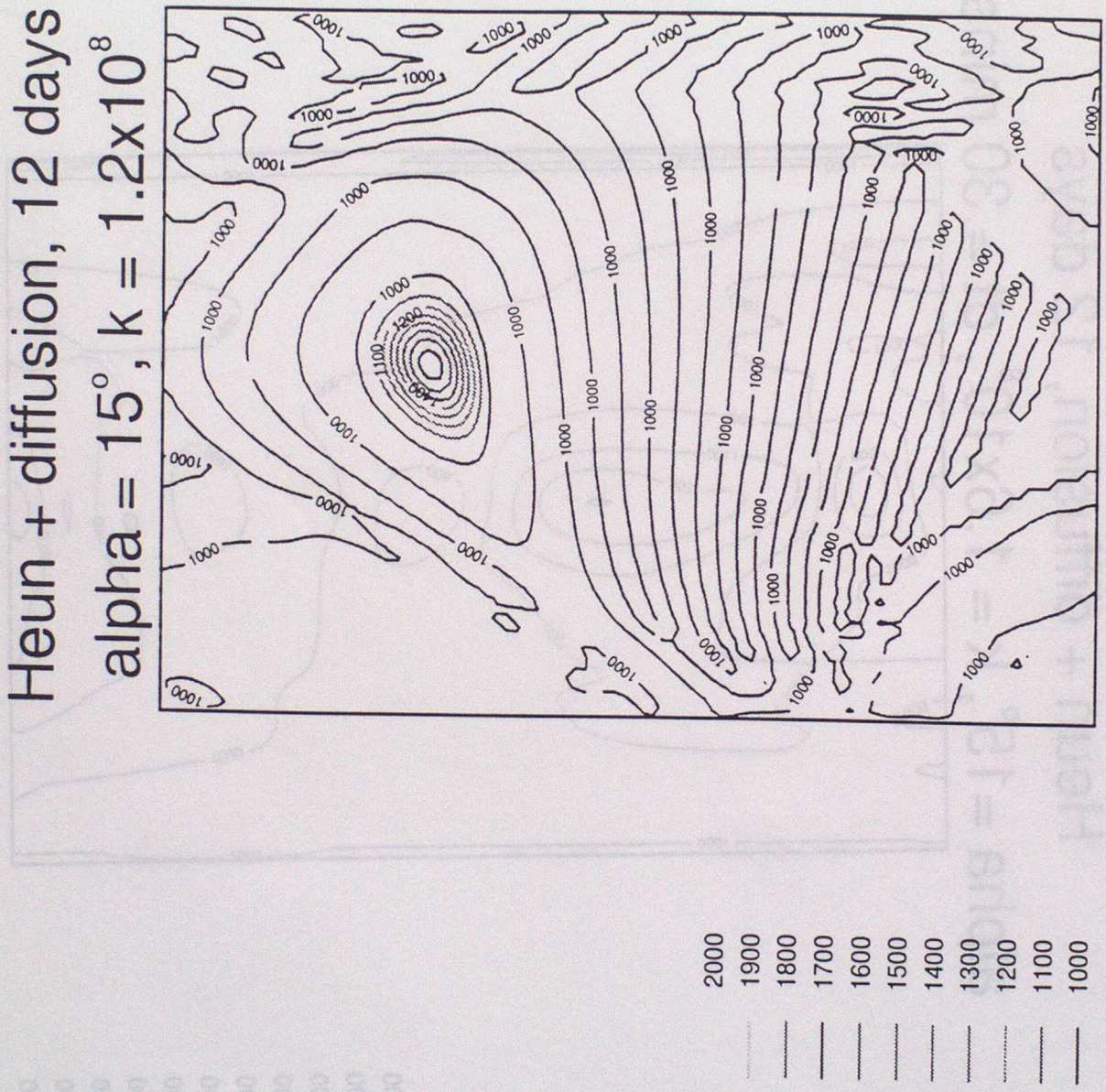


Figure 9: Heun scheme with filtering and 4'th order diffusion, time step = 450 seconds, $\alpha = 15°$, $k = 1.2x10^8$, 96 x 73 grid, result after 12 days.

Figure 10: TVD scheme , Superbee limiter, time step = 1800 seconds, $\alpha = 15°$, 96 x 73 grid, result after 12 days.

Figure 11: TVD scheme , Superbee limiter, time step = 1800 seconds, $\alpha = 90°$, 96 x 73 grid, result after 6 days.

Figure 12: l_2 error vs diffusion coefficient, $\alpha = 0$, time step = 1800 seconds, 4'th order diffusion, 96 x 73 grid.

Figure 13: l_2 error vs diffusion coefficient, $\alpha = 15°$, time step = 1800 seconds, 4'th order diffusion, 96 x 73 grid.

Figure 14: l_2 error vs diffusion coefficient, $\alpha = 0$, time step = 1800 seconds, 6'th order diffusion, 96 x 73 grid.

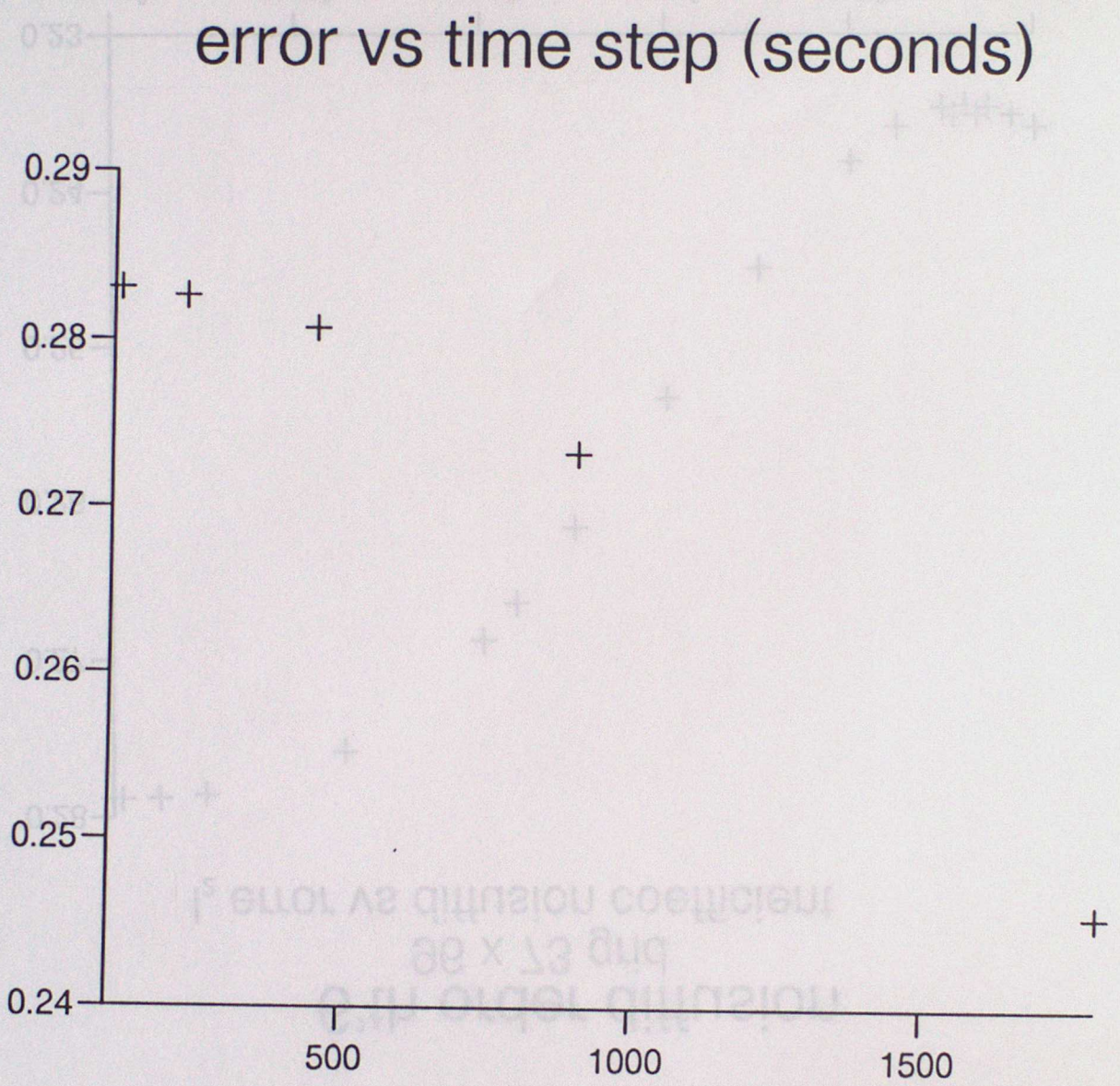


Figure 15: l_2 error vs time step, 96 x 73 grid, $\alpha = 0$, 4'th order diffusion.

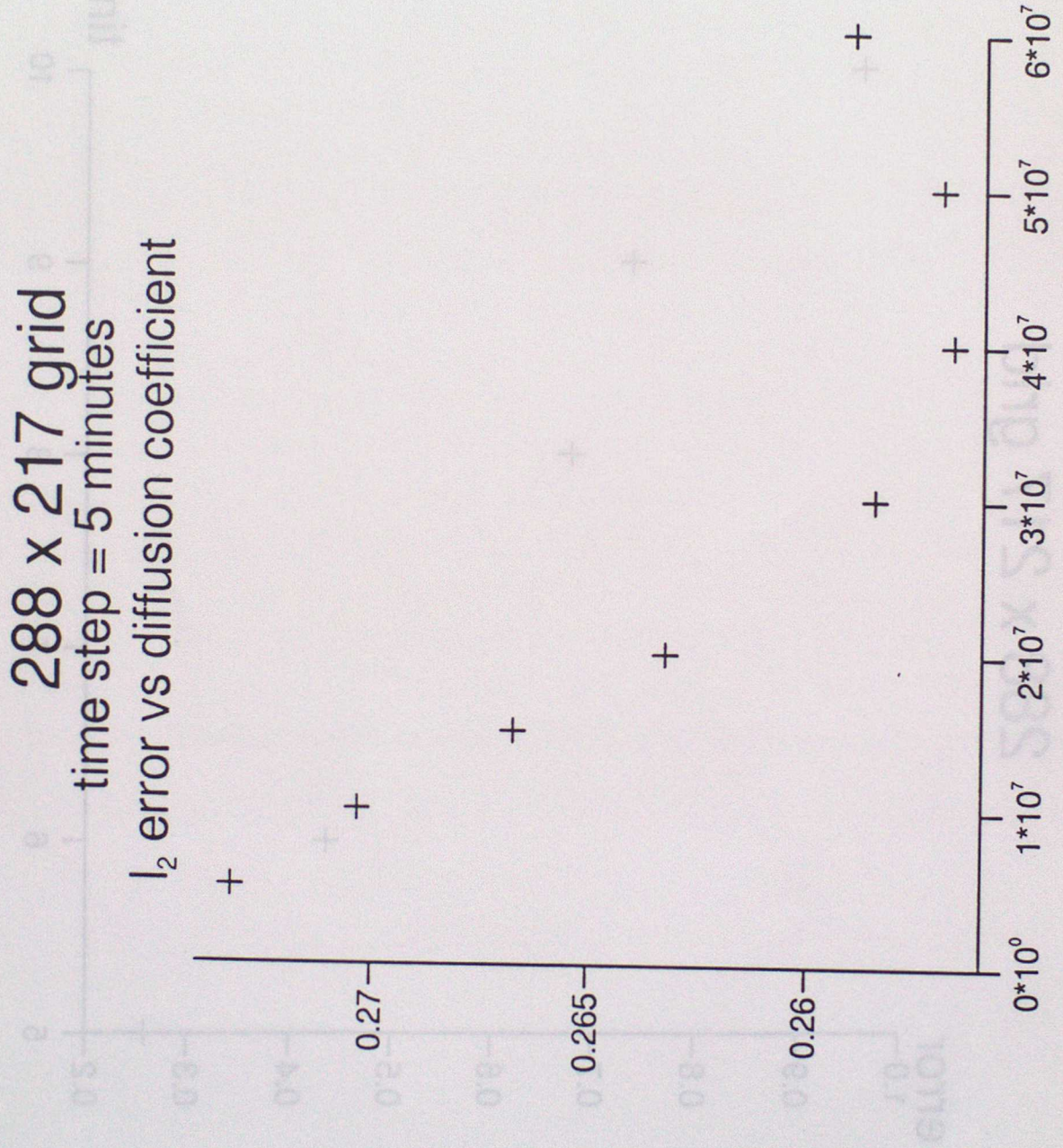


Figure 16: l_2 error vs diffusion coefficient, $\alpha = 0$, time step = 300 seconds, 4'th order diffusion, 288 x 217 grid.

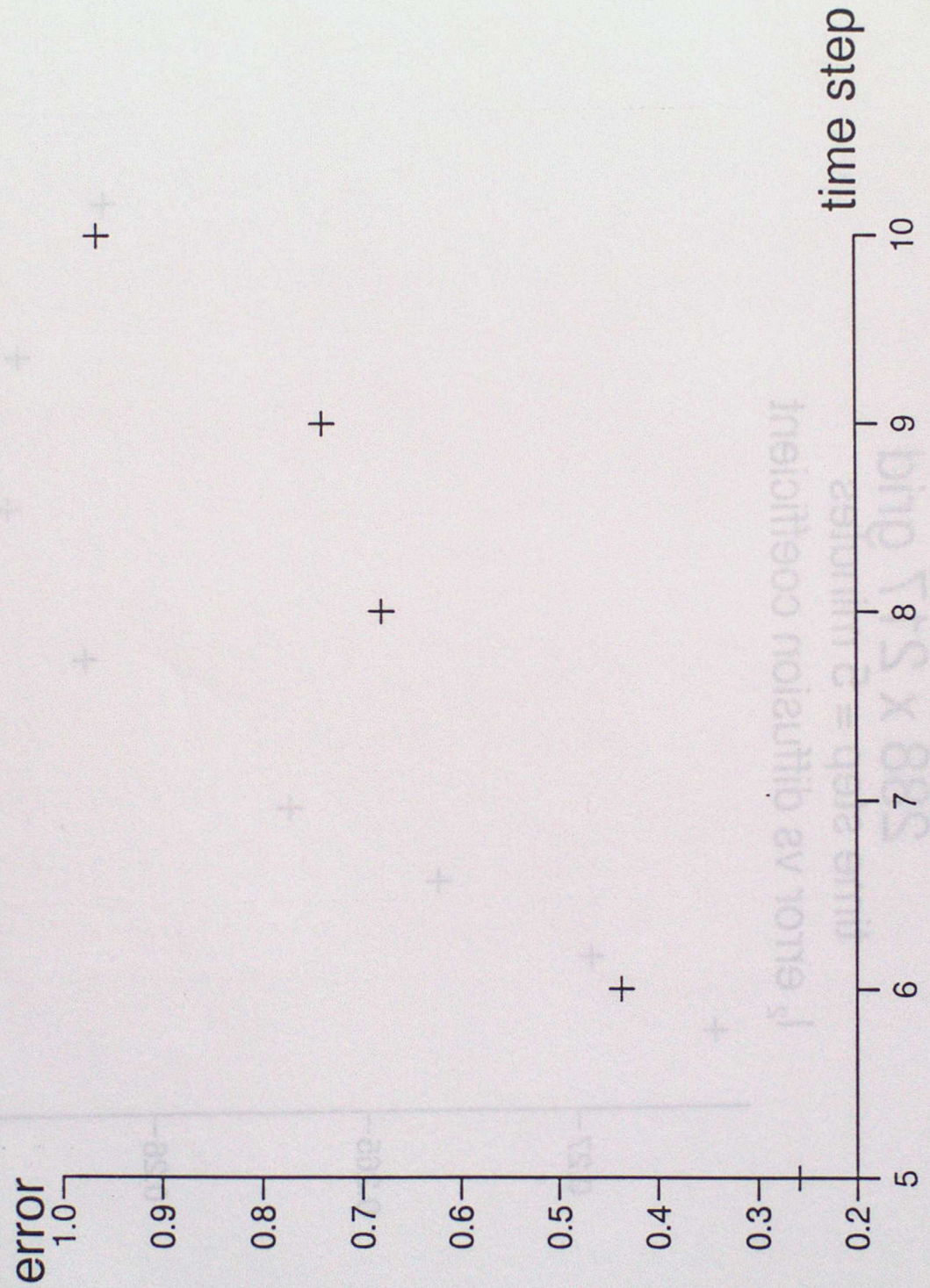


Figure 17: l_2 error vs time step, 288 x 217 grid, $\alpha = 0$, 4'th order diffusion.

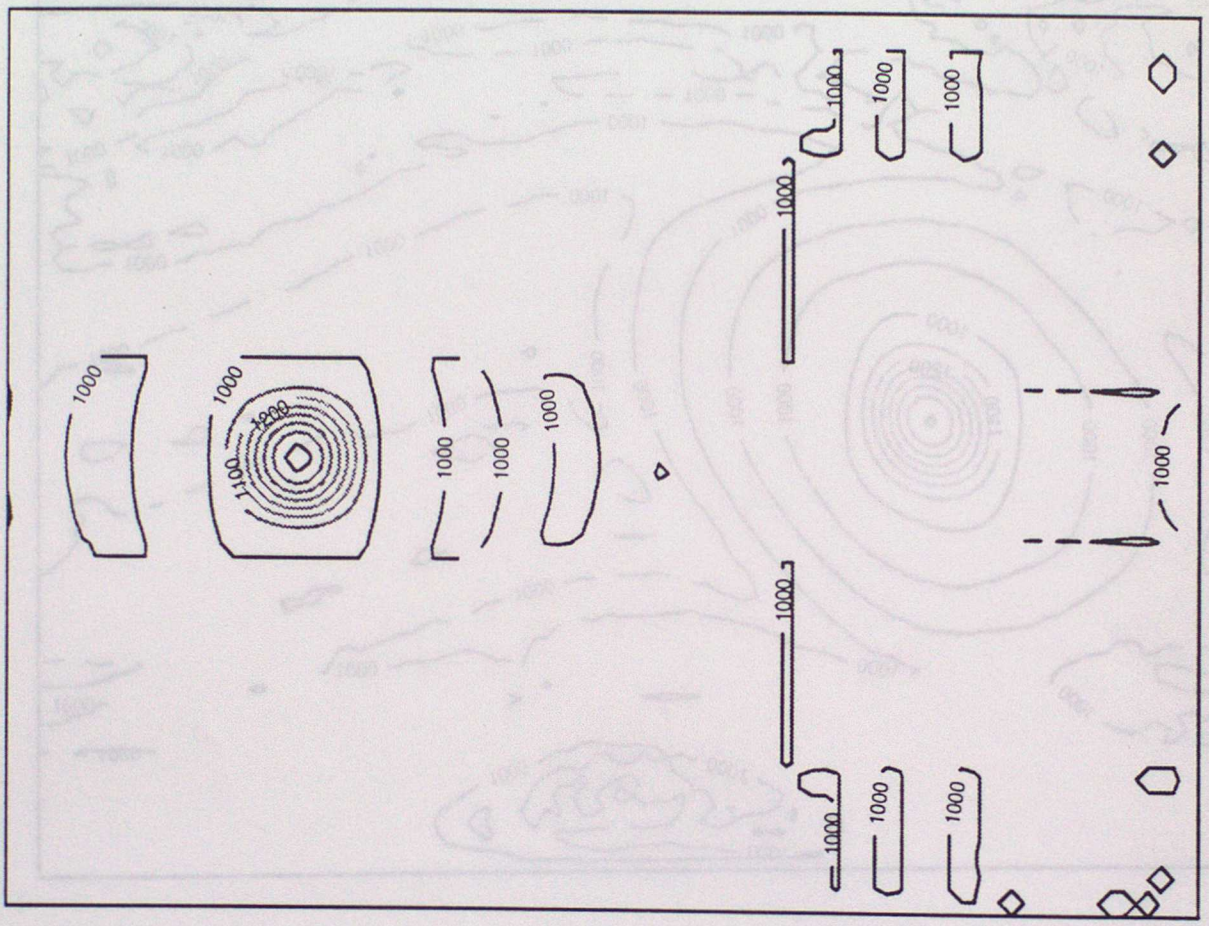


Figure 18: Semi-lagrangian code, 96 x 73 grid. $\alpha = 0$, time step = 1800 seconds.

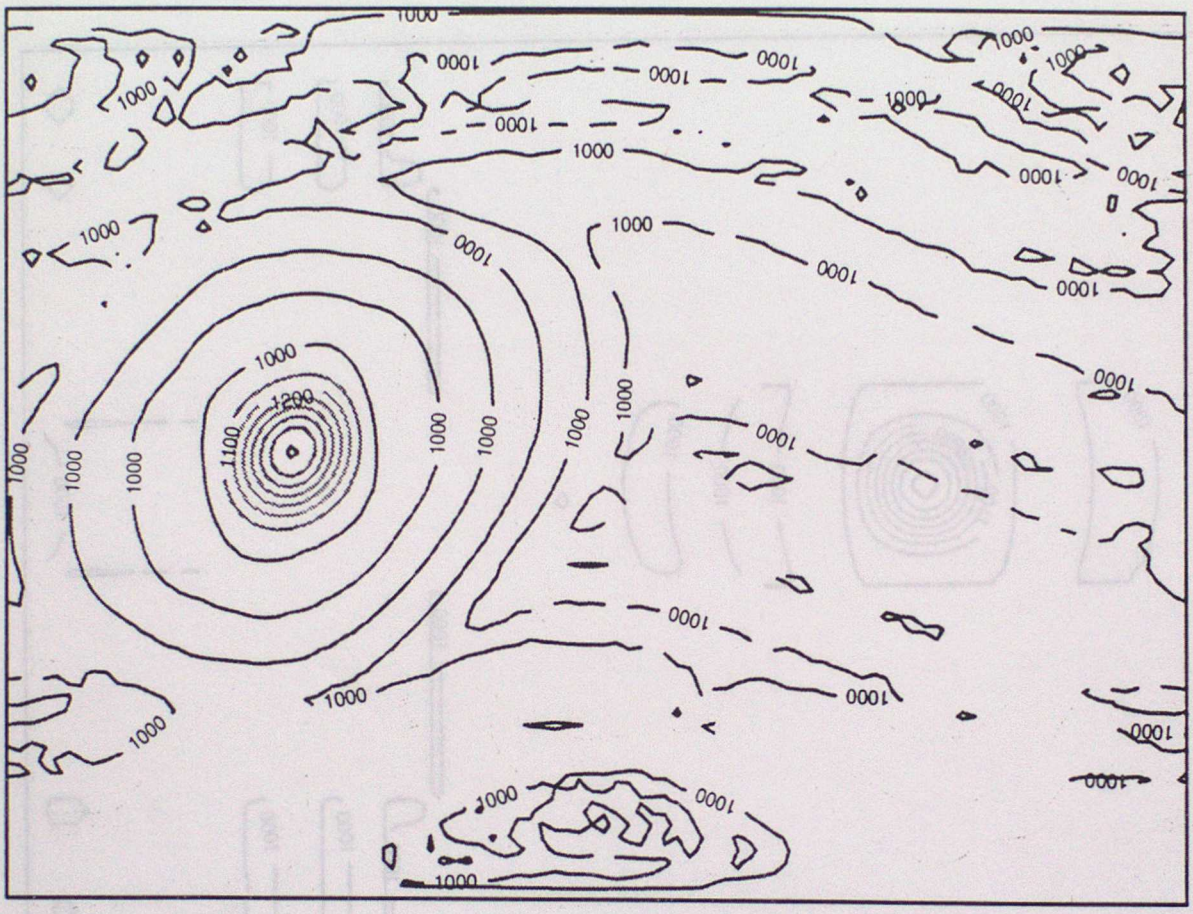
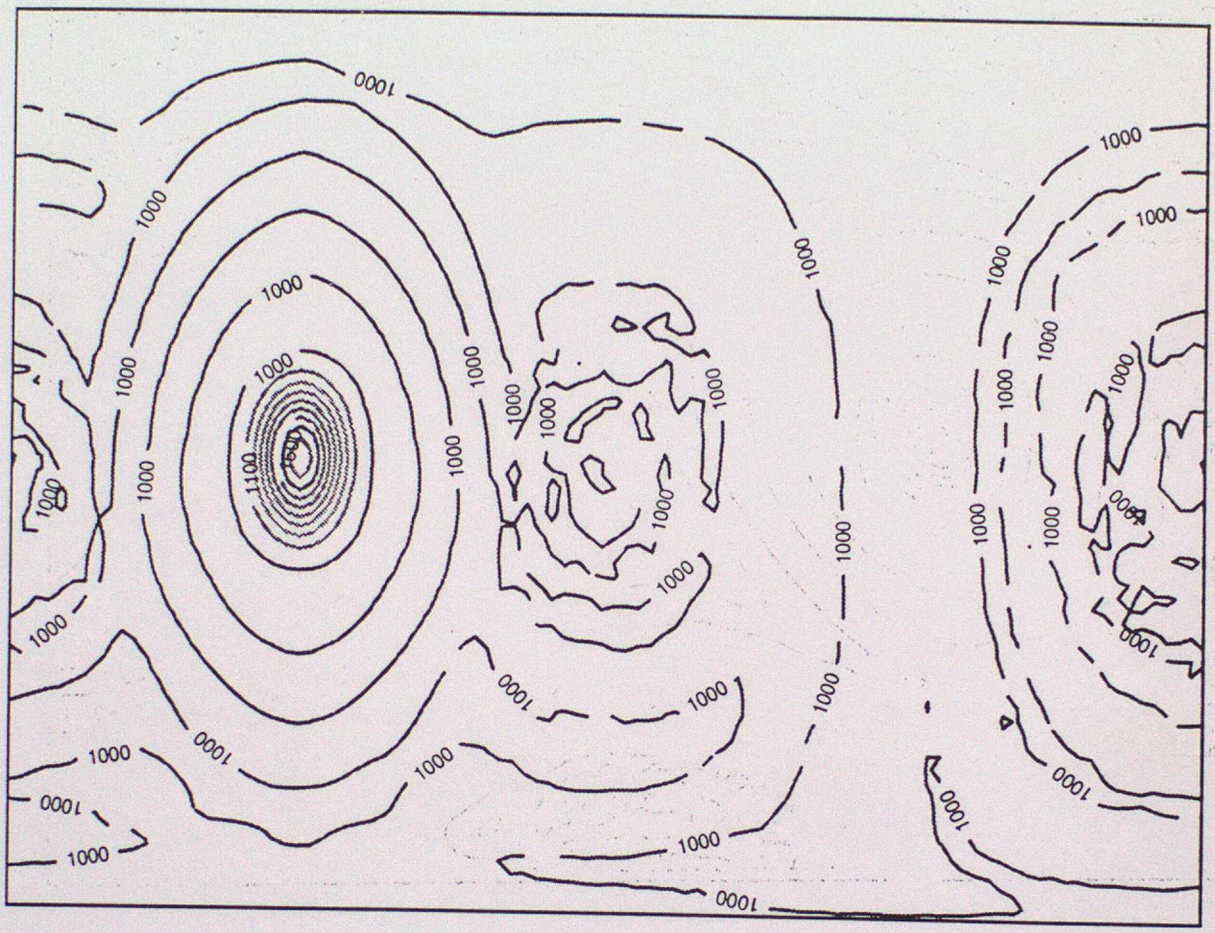


Figure 19: Semi-lagrangian code, 96 x 73 grid, $\alpha = 15°$, time step = 1800 seconds.



2000
1900
1800
1700
1600
1500
1400
1300
1200
1100
1000

Figure 20: Semi-lagrangian code, 96 x 73 grid, $\alpha = 90°$, time step = 1800 seconds